

CLAIMS

Sub 21
1. A method for programming a memory array, the method using programming pulses, the method comprising the step of:

adapting said programming pulses to the current state of said memory
5 array.

2. A method according to claim 1 wherein said step of adapting includes the steps of:

determining the voltage level of the programming pulse used to program generally the fastest bit of said memory array; and

10 setting an initial programming pulse level of said memory array to a level in the general vicinity of said programming pulse level of said generally fastest bit.

3. A method according to claim 2 and wherein said step of determining includes the steps of:

15 programming a small set of bits of said memory array;

setting a starting programming pulse level to a programming pulse level not higher than the programming pulse level used to program a fast bit of said -- small set;

programming generally all of the bits of said memory array beginning
20 at said starting programming pulse level; and

setting said initial programming pulse level to a programming pulse level in the general vicinity of a programming pulse level of a fast bit of generally all of said bits.

4. A method according to claim 1 wherein said general vicinity is not higher than said programming pulse level of said fast bit.

5. A method according to claim 3 wherein said generally all of the bits is all the bits of the array but the bits of said small set.

5 6. A method according to claim 3 wherein said generally all of the bits is all of the bits of said array

7. A method according to claim 1 and wherein said step of adapting includes the steps of:

measuring the current threshold level of a bit to within a predetermined
10 range; and

selecting an incremental voltage level of a next programming pulse for said bit in accordance with said measured current threshold level.

8. A method according to claim 7 and wherein said step of measuring includes the step of having multiple verify levels for said array.

15 9. A method according to claim 8 and wherein said step of measuring also includes the step of after a programming pulse, comparing a threshold level of a group of bits which have received said programming pulse to at least one of said verify levels and said step of selecting includes the step of selecting a next programming pulse level according to generally the highest verify level
20 achieved by said group.

10. A method according to claim 9 and also comprising the steps of removing any bit which has been programmed from said group and repeating said steps of comparing and selecting until there are no more bits in said group.

11. A method according to claim 8 and wherein said step of measuring also includes the step of after a programming pulse, comparing a threshold level of a bit which received said programming pulse to at least one of said verify levels and said step of selecting includes the step of selecting a next
5 programming pulse level according to generally the highest verify level achieved by said bit.

12. A method according to claim 7 and wherein the magnitude of said incremental voltage level corresponds to said measured current threshold level such that, after programming with said incremental voltage level, said bit
10 generally is fully programmed.

13. A method according to claim 7 and wherein the magnitude of said incremental voltage level corresponds to said measured current threshold level such that, after programming with said incremental voltage level, said bit generally is slightly less than fully programmed.

14. A method according to claim 13 and also comprising the step of final
15 programming said bit with a small incremental voltage level after said step of programming with said incremental voltage level.

15. A method for erasing a memory array, the method using erase pulses, the method comprising the step of:

20 adapting said erase pulses to the current state of said memory array.

16. A method according to claim 15 wherein said step of adapting includes the steps of:

determining erase conditions of the erase pulse used to erase a representative portion of said memory array; and

setting initial erase conditions of said memory array to the general vicinity of said erase conditions of said representative portion.

17. A method according to claim 15 and wherein said step of adapting includes the steps of:

5 measuring the current threshold level of a bit to within a predetermined range; and

 selecting an incremental voltage level of a next erase pulse for said bit in accordance with said measured current threshold level.

18. A method according to claim 17 and wherein said step of measuring
10 includes the step of having multiple verify levels for said array.

19. A method according to claim 17 and wherein said step of measuring also includes the step of after an erase pulse, comparing a threshold level of a group of bits which have received said erase pulse to at least one of said verify levels and said step of selecting includes the step of selecting a next erase pulse level
15 according to generally the lowest verify level achieved by said group.

20. A method according to claim 19 and also comprising the steps of removing any bit which has been erased from said group and repeating said steps of comparing and selecting until there are no more bits in said group.

21. A method according to claim 17 and wherein said step of measuring also
20 includes the step of after an erase pulse, comparing a threshold level of a bit which received said erase pulse to at least one of said verify levels and said step of setting includes the step of selecting a next erase pulse level according to generally the lowest verify level achieved by said bit.

to claim 1
e followin
ion and any

1911	1912	1913	1914	1915	1916	1917	1918	1919	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------